ICLICKER QUESTION

Which operation is faster on a Linked List than on an Array?

A. Get (an element)
B. Insert (in the middle)
C. Add (to the end)
D. None
Structuring Music

Copy

- Two types of copy for objects:
  - Shallow Copy
    - Copy only the reference to the object
    - Two references to the same object
  - Deep Copy
    - Copy the actual data

Deep Copy

```java
private static class Node {

  //...

  public Node copy() {
    return new Node(note.copy(), next);
  }
}
```

Repeat

```java
public void repeat() {
  Node head2 = head.copy();
  Node cur = head;
  Node cur2 = head2;
  while (cur.next != null) {
    cur2.next = cur.next.copy();
    cur = cur.next;
    cur2 = cur2.next;
  }
  cur.next = head2;
}
```
Consider the following two data structures:

- n integers stored in an array (of length n)
- n integers stored in a linked list (with n nodes)

Which requires more space in memory?

A. The array.
B. The linked list.
C. They both require the same.

### SIZE COMPARISON: N INTS

- The array requires
  - Reference... 4 bytes
  - (int... 4 bytes) * n
  - = 4n + 4
- The linked list requires
  - Reference... 4 bytes
  - (int... 4 bytes + Reference... 4 bytes) * n
  - = 8n + 4
- Of course, this overhead becomes relatively smaller with larger data types.

### SIZE COMPARISON: N OBJECTS

- Say we have an object of size 64 bytes
- The array requires
  - Reference... 4 bytes
  - (Object... 64 bytes) * n
  - = 64n + 4
- The linked list requires
  - Reference... 4 bytes
  - (Object... 64 bytes + Reference... 4 bytes) * n
  - = 68n + 4